SELF-TESTING SOC WITH REDUCED MEMORY REQUIREMENTS AND MINIMIZED HARDWARE OVERHEAD

Ondřej NOVÁK[*], Zdeněk PLÍVA[**], Jiří JENÍČEK[**], Zbyněk MADER[**], Michal JARKOVSKÝ[**]
[*]Czech Technical University in Prague, Dept. of Computer Science & Engineering,
Karlovo nám. 13, Praha 2,Czech Republic, e-mail: novako3@fel.cvut.cz
[**]Technical University of Liberec, dept. ITE, Hálkova 6, 461 17 Liberec I, Czech Republic,
e-mail: zdenek.pliva@tul.cz, jiri.jenicek@tul.cz, zbynek.mader@tul.cz, michal.jarkovsky@tul.cz

ABSTRACT

This paper describes a methodology of creating a built-in diagnostic system of a System on Chip and experimental results of the system application on the AT94K FPSLIC with cores designed according to the IEEE 1500 standard. The system spares memory and keeps acceptable test access mechanism requirements. The diagnostic system uses a built-in processor for test control, the embedded RAM memory for storing both the compressed test vectors and the partial reconfiguration bit streams and the FPGA (Field-programmable gate array) part of the chip for the wrapped cores implementation. The highly compressed test vectors are transferred from the memory to those selected cores that are reconfigured into the embedded tester cores. The patterns are decompressed within the internal scan chains of the embedded tester cores and they are simultaneously fed into the parallel scan chains of the cores under test through Test Access Mechanism (TAM) and standard wrappers. After having tested the first cores under test the TAM of the System on Chip (SoC) is partially reconfigured with the help of the partial reconfiguration bitstreams stored in the RAM memory and the till now untested cores are tested by those cores that start to serve as embedded testers. By this traveling reconfiguration and testing the whole circuit can be tested. For test data compression we use a test pattern compaction and compression algorithm called COMPAS. It reorders and compresses test patterns previously generated in an Automatic Test Pattern Generation (ATPG) in such a way that they are well suited for decompression by the scan chains in the embedded tester cores. The algorithm compresses the test patterns by overlapping patterns originally generated by an ATPG. The volume of test data stored in the embedded RAM is substantially lower than the compacted ATPG test data that are compressed by other compression method. The COMPAS algorithm spares the CPU time and CPU memory requirements; both of these parameters are linearly dependent with the complexity of the tested core.

Keywords: *SOC testing; mixed-mode testing; test pattern compression; dynamic FPGA reconfiguration*

1. INTRODUCTION

This paper describes our research and experimental results in the field of testing the complex circuits designed according the IEEE 1500 standard [13]. Testing of complex SoC circuits has to overcome the following challenges:

- To reach better test quality than can be obtained by pseudorandom test set application
- To reduce the tester memory requirements
- To reduce the amount of data transferred to/from the tested chip
- To keep the test time short
- To keep the hardware overhead acceptably low.

The paper describes methods and experimental results obtained by solving the challenges mentioned above. The following subsections of the Introduction describe the problems concerned with the built-in diagnostic system and their solutions. The second section describes the results we have obtained in the field of test pattern compression. Section 3 addresses the technical solution of the experimental diagnostic system built-in the AT94K FPSLIC and Section 4 concludes the paper.

1.1. Test pattern quality

Built-in pseudorandom or weighted random testing can be a solution of the problems with the memory for storing deterministic test patterns but still there remain random resistant faults, which have to be tested from an Automatic Test Equipment (ATE) with deterministic patterns. Mixed-mode testing uses built-in pseudorandom pattern generators, which are usually used for generating first several thousands of test patterns (typically 10 000 patterns) and deterministic patterns are applied after the pseudorandom testing phase in order to test the random resistant faults. The deterministic patterns can be compressed, the decompression is usually done in the same automaton as it was used for generation of pseudorandom test sequence; the seeds are stored in an ATE. Linear feedback shift register (LFSR) reseeding methods [17] assume that a large portion of bits in the test patterns are unspecified. The on-chip LFSR is seeded with seeds that guarantee that the bit sequence generated by the LFSR matches the deterministic patterns at the specified positions. The number of bits stored in a tester memory is relatively small but the total number of clock cycles, which is needed for testing, may be high. Random part of mixed-mode test is time and energy consuming.

The usefulness of a test compression method is influenced not only by the compression ratio but also by the complexity of the decompressing automaton and by the computational complexity of the algorithm for finding the compressed test sequence.

In order to keep the decompressing hardware minimal it is possible to compress test patterns by overlapping the patterns that are serially shifted into scan chains (SC). If the SC do not contain internal flip-flops that can be rewritten by the test responses, the patterns can be decompressed during the test session with no additional hardware by simple performing one or more SC shifts.

This approach was firstly described in [9] and later in [31]. The test pattern compression uses an algorithm for finding contiguous and consecutive maximally overlapping scan chain vectors for the actual scan chain vector. These vectors are checked whether they match with one or more of test patterns, which were previously generated and compacted with the help of some ATPG and which were not employed in the scan chain sequence yet. In [24] we presented an improved algorithm, which speeds up the computation of the overlapped patterns by searching for the successors of the all zero seed only and which improves the compression efficiency by fault simulation, which is performed during the phase of finding the overlapped patterns. The fault simulation enables the algorithm to skip the patterns testing the already covered faults. The algorithm compresses test vectors with don't care bits; each pattern covers one fault only. The algorithm was implemented in the COMPAS (COMpressed test PAttern Sequencer) software tool [14]. COMPAS is intended to be used for preparation of test sequences of cores under test (CUT) that are equipped with a scan chain at least on the inputs.

1.2. Test Access Mechanism (TAM)

A test set can be controlled by a tester or by a BIST controller. It could be advantageous to use an embedded processor instead of a specialized controller with a RAM. As the RAM size is limited, the test set has to be as small as possible. Further testing speed improvement could be obtained by minimizing the amount of data transferred between the processor and the tested cores. From this reason it is worthwhile to send the compressed data from the processor to the decoders that are placed closely to the tested cores and to leave the decoders to decode the patterns independently on the processor activity. This arrangement can speed up testing as the clock frequency of the core flip-flops could be higher than the processor clock frequency and the processor can prepare next data during decoding the previous pattern. Another problem arises when using cores with the SC that contain internal flip-flops; if we have to guarantee not corrupting test patterns by CUT responses and simultaneously catching all test responses we have to scan in and scan out the whole test pattern after each system clock application. The RESPIN (Reusing Scan Chains for Test Decompression) test architecture [10] solves both pattern decompression and reducing the data traffic between tester and CUT.

This architecture reuses scan chains of different cores for updating the tested core scan chain content (Fig. 1). The RESPIN architecture temporarily divides the circuit into the core under test (CUT) and the embedded tester core (ETC). The data transfer mechanism between the tester and ETC can be denoted as a narrow TAM as the demanded transfer capacity is low. The TAM between the ETC and CUT is wide as the data transfer is done parallel and on a higher clock frequency. The CUT and ETC have several parallel scan chains. The ETC chains are concatenated into a serial scan chain; a feedback tap connects the ETC last chain output with the first bit input through a multiplexer. According the multiplexer control input, ETC can either load a bit from the tester or shift the scan chain circularly. The parallel chains of the CUT are connected with the parallel ETC chain outputs. This test pattern updating mechanism guarantees that the patterns, which are shifted through a CUT SC during several test steps, are not mixed with the CUT responses. An additional multiple input signature register (MISR) connected to the SC outputs can be exploited for capturing all the test responses. The conditions for effective testing are: the ETC has at least the same number of chains as the CUT; the CUT chains are not longer than the corresponding ETC chains and the number of scan cells of the CUT and the total number of ETC scan cells incremented by one have not a common divider. If it is not possible to find an ETC core that fulfils the above mentioned conditions, more than one core can be used for creating the ETC.

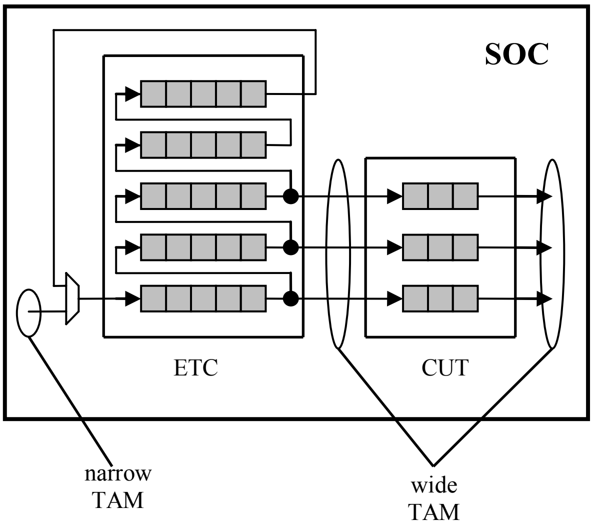

Fig. 1 ETC and CUT in the RESPIN architecture

A number of TAM architectures have been proposed and published [8]. The basic solutions are:

- Architecture with multiplexing
- Distributed architecture.

Various combinations of those architectures may co-exist. The architecture called TestBus [15] was developed from the architecture with multiplexing and distributed architecture by their combination. Another possible architecture is TestRail [21] which tries to combine strengths both of the test bus and boundary scan test. In this paper we propose to use a partial reconfiguration for switching the diagnostic busses instead of the classical architectures.

As the IEEE 1500 standard provides a standardized possibility of SoC core testing, every core designer should try to use the normalized wrapper. Such wrapper has a mandatory Wrapper Serial Port (WSP) port. If the test system contains hardware blocks such as random test generator, decompressor or response analyzer, all of blocks must have a WSP port to synchronize tests. Such control mechanism must be created for different solutions of test systems as well. For this reason, we will not take the hardware overhead of the WSP port into account for hardware overhead comparisons, as it consumes approximately the same die area in all designs.

1.3. Reconfiguration

With the advent of programmable hardware we are able to provide the same possibility as with software – upgrades and updates in situ, in the embedded system without removing the reconfigured part. In case of FPGAs the final functionality of the circuit depends on the configuration bitstream loaded into the device from external memory. The novel FPGA circuits are dynamically reconfigurable at runtime. These dynamically reconfigurable FPGA circuits (D_FPGA) have a capability to change the behaviour of one part of the circuit; the rest part is fully operational without changes and without interruption. Generally, each memory-based FPGA can be reconfigured dynamically. In the currently known dynamically reconfigurable devices two techniques are used: "partial configuration" and "Multiple-context configuration memory" [29].

Partial configuration allows selective access to the configuration memory, the speed of dynamic reconfiguration is proportional to the number of configuration memory locations which need to be changed; this technique is used in e.g. Xilinx XC6200 or Atmel AT6000. Multiple-context configuration memory maps successive configurations in multiple contexts of the configuration memory. The dynamic reconfiguration is performed by swapping a selected inactive configuration memory context into the active context. Also both of these techniques can be combined on one device. Nowadays, the dynamically reconfigurable devices commercially accessible are Virtex family by Xilinx [32] and AT40K and AT94K families by Atmel [2]. The use of reconfigurable design is not a quite new idea; such devices are available for a few years. However, the use is mostly restricted to academic or research applications. There are two main reasons why this useful technology is not widespread in the common applications:

- Lack of tools which can easily and efficiently manage the design of reconfigurable application.
- Lack of reconfigurable applications worthy of consideration (or lack of the spunk).

Reconfiguration of the TAM for a SoC testing seems to be an efficient exploitation of the partial reconfiguration capability of FPGAs. As the Atmel FPGAs can efficiently perform the fine grained reconfiguration we decided to use it for an implementation of the self-testable SoC (System on Chip) design. The diagnostic system uses RESPIN architecture which is based on the IEEE 1500 standard. The partial reconfiguration is used for connection among ETCs, CUT and the feedback multiplexer.

The main advantage of the proposed solution is that all the reconfiguration bitstreams are stored inside the chip. Thereafter the reconfiguration process can be controlled by the embedded processor and the only communication between the tested SoC and the external test supervisor is a request for execution the test and checking the results of the done tests.

2. TEST PATTERN COMPACTION AND COMPRESSION

The self-testing system uses compressed patterns that are prepared offline by the COMPAS algorithm [24], [25]. This algorithm was speeded up in order to be able to handle with greater cores in acceptable CPU time. In this section we describe the main principles of the improved COMPAS version. The algorithm minimizes the overlapped test sequence length in which all test patterns that detect at least one fault undetected by resting patterns appears at least once in the SC. The test sequence has to be decompressed in the ETC cores so that each pattern can be applied in the CUT with several parallel cores.

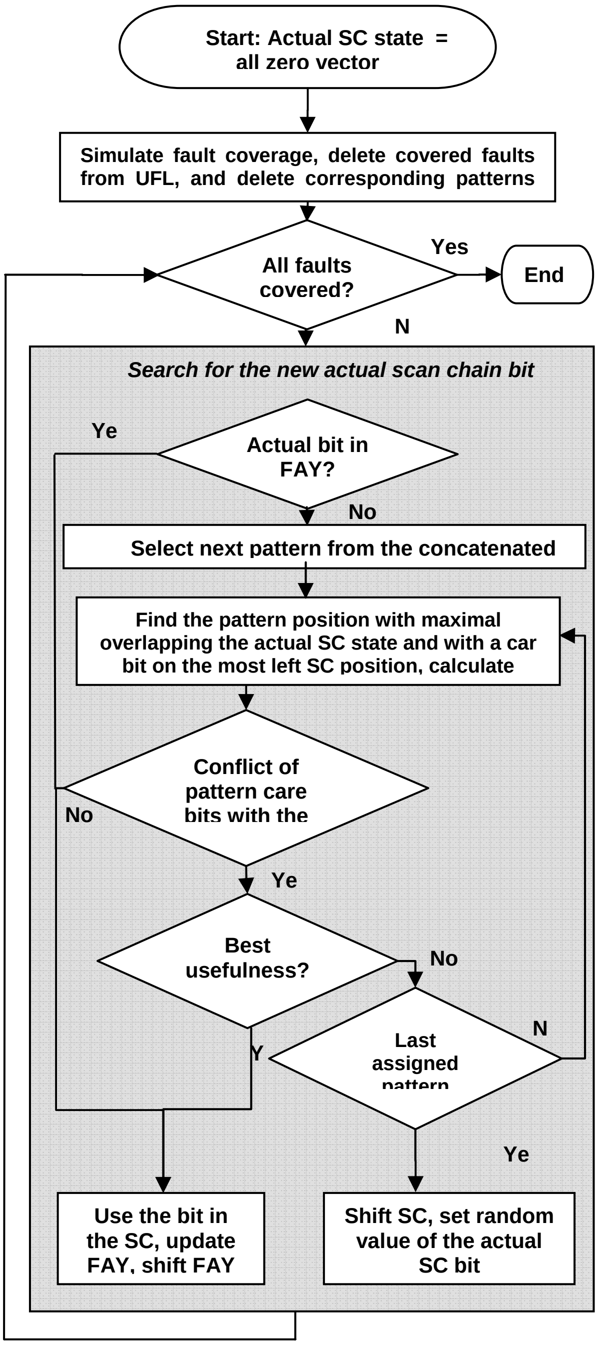
Start: Actual SC state = all zero vector
Simulate fault coverage, delete covered faults from UFL, and delete corresponding patterns
All faults covered?
Yes
End
N
Search for the new actual scan chain bit
Ye
Actual bit in FAY?
No
Select next pattern from the concatenated
Find the pattern position with maximal overlapping the actual SC state and with a car bit on the most left SC position, calculate
Conflict of pattern care bits with the
No
Ye
Best usefulness?
No
N
Y
Last assigned pattern
Ye
Use the bit in the SC, update FAY, shift FAY
Shift SC, set random value of the actual SC bit

Fig. 2 The main loop of the algorithm of finding the ATE memory bits

At the beginning the Test Pattern List (TPL) - prepared by the ATPG together with the corresponding Undetected Fault List (UFL) is used. Test vectors using three values algebra – t €(0,1,X), where X means that don't care value has to be generated for the tested circuit. An ATPG tool that enables generating non-compacted test patterns for each considered fault has to be used. In this way we can distinguish, which pattern belongs to which fault.

The main loop of the finding the compressed test sequence algorithm is described in Fig. 2. Let us suppose that the SCs are reset before testing, which means that the all zero pattern is considered to be used as the first one. The fault coverage of this pattern is simulated and the detected faults are deleted from the UFL, test patterns corresponding to the detected faults are deleted from the TPL. Then the algorithm tries to compact the test set by overlapping resting patterns with the actual vector. The algorithm finds, whether the logic value 0 or logic value 1 is better to be used as the next most left chain bit. To do this the algorithm finds positions of all patterns, in which the actual chain bits maximally overlap the pattern and for which the actual bit to be introduced into the scan chain has not a don't care value. Simultaneously the algorithm determines for how many future clock cycles of the SC it is not necessary to recalculate the position of the pattern.

This information is used for skipping pattern recalculation for cases when don't care bit groups are present in the patterns. Skipping pattern position and usefulness recalculation is enabled by using a concatenated list of pattern pointers. The pointers indicate the position of the SC where the pattern can eventually influence the SC content. Chaining the pointers reduces the complexity of enumerating the absolute position. After finding the position the algorithm has to count the usefulness of the treated pattern. The usefulness is characterized by the number of don't care bits and the possibility of overlapping the pattern. It is computed according the formula given in [25]. An example of positioning the vectors is given in Tab. 1. Vector 1 overlaps the SC in two bits but it has a don`t care bit on the actual position. This don`t care bit is followed out by 2 other don`t care bits. This means that this vector will not be evaluated for two next algorithm cycles.

The usefulness criterion prefers patterns that have high number of care bits and simultaneously that have maximum number of the care bits overlapped with the scan chain. Then the algorithm compares the number of the most useful patterns with logic value 1 on the actual position and the number of patterns with logic value 0 on this position. If the number of ones is greater than the number of zeros the input actual bit is fixed to logic value 1, in the other case to logic value 0. This way of setting the actual bit guarantees that a maximum number of the most useful patterns could be encoded.

When searching for the most useful pattern it checks whether the exercised pattern matches with bits, which will be necessary to be generated in the future clock cycles because of some previously selected patterns.

If some position of FAY is reserved for a logic value that is clashing with the exercised pattern bit value, the algorithm compares the usefulness of both patterns and the winner is used in the future content of the FAY, the other bit is deleted from the FAY but the corresponding pattern is kept in the TPL. The fault simulation is performed and the faults and patterns, which correspond to the covered faults, are removed from the lists. This sequence guarantees that a test pattern was deleted from the TPL either because it was definitely placed into the generated sequence or the corresponding fault was tested by some other test pattern. If there are not remaining faults in the UFL the algorithm is finished.

Table 1 Positions of 4 vectors that overlap the SC

	future bits					actual bit	generated bits					
clock cycle	**5**	**4**	**3**	**2**	**1**	**0**						
SC content						**?**	**0**	**0**	**1**	**1**	**1**	**0**
vector 1			**1**	**X**	**X**	**X**	**0**	**0**				
vector 2			**1**	**X**	**X**	**1**	**0**	**0**				
vector 3					**X**	**1**	**X**	**0**	**1**	**1**		
vector 4	**0**	**X**	**1**	**1**	**1**	**1**						

2.1. System COMPAS

The COMPAS can be remotely run on the web site [14]**.** In the current version of the COMPAS we use test patterns generated by the Atalanta ATPG tool [18] and the Hope fault simulation tool [19] is used for fault simulation. After accelerating the algorithm by using concatenated pointer lists enabling omitting recalculations of the patterns with don't care bits the CPU time for pattern compression is proportional to the circuit complexity. There are more than 99 % don't care bits within the test pattern bits of the larger circuits from ISCAS89 and ITC99 benchmark set. However, care to don't care bits ratio is strongly dependent on both circuit structure and ATPG quality. For the core of the benchmark circuit s38417 the CPU time was 291 s. on the PC with Intel Pentium IV, 2,8 MHz machine. This fact is illustrated in the graphs in Fig. 3, where we plot the CPU time (given in seconds) against the number of core gates and number of generated bits in the compressed test sequence for different ITC and ISCAS benchmark circuits [5]. The memory requirements for COMPAS are very low as COMPAS works with precompressed test data. ATPG output containing lot of don't care bits is stored in the memory as sparse vectors, storing only values of care bits and their positions. It has been proved that in different computation phases it is faster to use original decompressed test vector instead of sparse ones, therefore small cache containing decompressed vector has been created. Cache hit ratio is better than 60% in all up to now checked circuits.

In addition two pointers are stored for each test pattern for don't care skipping. After detecting a fault the corresponding pattern and all its temporary structures (like pointer lists) are removed from the memory, so the memory requirements decrease during the computation. Considering the 32 bit width of the memory word the maximum consumed memory for support structures of a circuit with one million of faults is equal to about 11.5 MB. COMPAS runs both on the Unix and Windows platform.

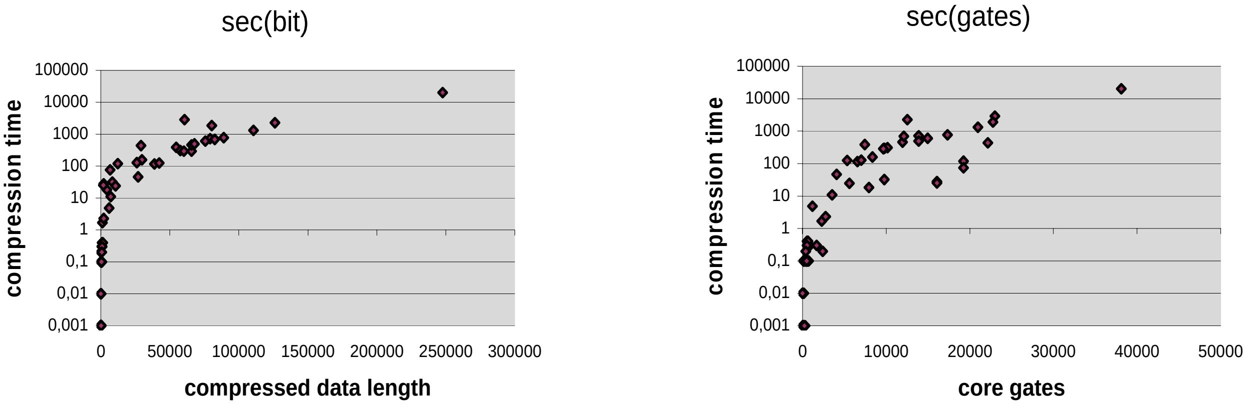

Fig. 3 CPU time (given in seconds) against the number of core gates and number of generated bits

2.2. Memory and CPU time requirements

In the following text, we compare the proposed method of compression and decompression with other compression and mixed-mode testing approaches.

Tab. 2 shows the resulting CPU time for finding the compressed test sequences and their lengths for ITC99 benchmark circuits [7], obtained by COMPAS. In Tab. 3 we have compared the numbers of stored bits of the greatest ISCAS circuits for some well known test pattern compression methods and for the proposed algorithm. This comparison was not possible to be done for more complex benchmark circuits as the results of other methods were not available. In the second column we plotted the test data volume for ATPG vectors, which were compacted only [4]. Next column shows the number of stored bits for statistical coding of the test patterns from the previous column [1]. The fourth column results correspond to a combination of statistical coding and LFSR reseeding [16]. The next columns summarize results of compression with parallel/serial scan chains [26] and frequency directed codes [6]. The results for the method of Embedded Deterministic Test are presented in the next column [28]. The column RESPIN++ shows the numbers of bits stored in the ATE for the RESPIN++ architecture given in [30].

Table 2 ITC benchmark test set lengths

circuit	CPU time (sec.)	# of bits	Time (sec.)/ 1000 bits
b14_1_C	116.0	38,702	2.997
b14_1_opt_C	46.3	26,943	1.718
b14_C	291.0	65,677	4.431
b14_opt_C	124.7	42,177	2.957
b15_1_C	2,270.6	126,077	18.010
b15_1_opt_C	386.0	54,697	7.057
b15_C	158.7	29,666	5.350
b15_opt_C	127.4	26,006	4.899
b17_1_opt_C	2,893.1	60,721	47.646
b17_opt_C	1,885.6	80,483	23.429
b20_1_C	714.5	79,159	9.026
b20_1_opt_C	312.2	57,585	5.422
b20_opt_C	461.2	65,726	7.017
b21_1_C	492.2	67,910	7.248
b21_1_opt_C	290.4	60,215	4.823
b21_opt_C	692.5	82,422	8.402
b22_1_C	1,333.2	110,602	12.054
b22_1_opt_C	606.3	75,738	8.005
b22_opt_C	769.7	88,888	8.659

We can see that the number of bits, which are stored in a memory, is substantially lower for the proposed method than for other pattern compressing methods. We have to note that a majority of the tabulated pattern compression methods do not use a fault simulation after encoding a new test pattern (with the exception of the method [30]). These methods use compacted test sequences, the fault coverage was simulated during test pattern generation in the ATPG during the process of pattern compaction. The total number of clock cycles for each method is given by the number of applied pseudorandom patterns, number of deterministic patterns and by the length of the longest parallel scan chain of the CUT. The test time will directly depend on this number. We have compared the obtained COMPAS test time with other mixed-mode testing methods and we have found that COMPAS provides substantially lower numbers of clock cycles than other methods while the numbers of necessarily stored bits are comparable.

Table 3 Comparison of lengths of compressed data for different methods

Circuit name	Min-Test [4]	Stat. Coding [1]	Reseeding [16]	Illinois Scan [26]	FDR Codes [6]	EDT [28]	RESPIN ++ [30]	COMPAS [25]
	# of bits	# of bits	# of bits	# of bits	# of bits	# of bits	# of bits	# of bits
s13207	163,100	52,741	11,285	109,772	30,880	10,585	26,004	4,402
s15850	58,656	49,163	12,438	32,758	26,000	9,805	32,226	8,433
s38417	113,152	172,216	34,767	96,269	93,466	31,458	89,132	21,280
s38584	161,040	128,046	29,397	96,056	77,812	18,568	63,232	6,825

Table 4 Comparison of memory requirements and time for test.

		folding counters		RESPIN		proposed	
circuit name	# inputs	# bits	# clk	# bits	# clk	# bits	# clk
c7552	207	9E3	4E6	-	-	5E3	1E6
c2670	233	5E3	4E6	-	-	4E3	8E5
s13207	700	25E3	2E7	4E3	2E8	4E3	3E6
s15850	611	18E3	2E7	6E3	2E8	7E3	4E6
s38417	1,664	136E3	2E8	3E4	5E9	2E4	3E7
s38584	1,464	33E3	6E7	6E3	2E8	6E3	9E6

2.3. Test Time

The total number of clock cycles for mixed mode test methods is given by the number of applied pseudorandom patterns, number of deterministic patterns and by the lc., where lc is the scan chain length. We have compared the proposed system test time with the method [11] which uses so called Folding Counters and the original RESPIN method [10] as an example. These methods use 10,000 random test patterns that are pattern by pattern serially shifted into the scan chain and than the random resistant faults are tested. In case of Folding Counters [11] the automaton is reseeded several times in order to pass through the states that correspond with the deterministic test vectors. In case of RESPIN the hard fault detecting deterministic patterns are encoded during two encoding phases into a RESPIN input sequence. In order to keep fair comparison we have considered that no pseudo input reduction is applied in all methods (no additional CUT dependent hardware is used inside the SC). In case of Folding Counters and the proposed method we considered that only one SC is in one core; in case of RESPIN the parallel SCs had the length of 100 bits. Tab. 4 shows the results of the comparison for several ISCAS circuits (some number are given in the exponential form (e.g. 2E3 = 2,000). We can see that COMPAS provides substantially lower numbers of clock cycles than other methods while the numbers of necessarily stored bits are comparable. Other not mentioned mixed-mode testing approaches have similar requirements on the number of clock cycles, so we may say that the proposed method is less time consuming than mixed-mode testing approaches.

3. EXPERIMENTAL SOC SELF-TESTING SYSTEM

The experimental diagnostic system was built on the FPSLICTM AT94K40AL circuit. It is a dynamically reconfigurable programmable SoC, which integrates Atmel SRAM, FPGA and an 8-bit AVR processor [2].

The communication between these main parts is entrusted to 8 bits bus and 16 internal interrupts (see datasheet). Beside these two main parts 36 kB SRAM (20 – 32 kB for program, the rest for data), UART and JTAG interface [12], watchdog timer, two counters etc. are placed on the chip. The user interface allows loading the AVR control program and initial bitstream from a PC to the RAM through the serial port. The on-chip AVR manages also the reconfiguration process. The AVR has a direct access to the FPGA's configuration memory and it is able to upload the configuration bitstreams. The data stored in the FPGAD register are used for programming the FPGA configuration SRAM cells according the address from the registers FPGAX, FPGAY FPGAZ. The content of FPGAX/Y/Z/D registers is set by AVR (see Fig. 4).

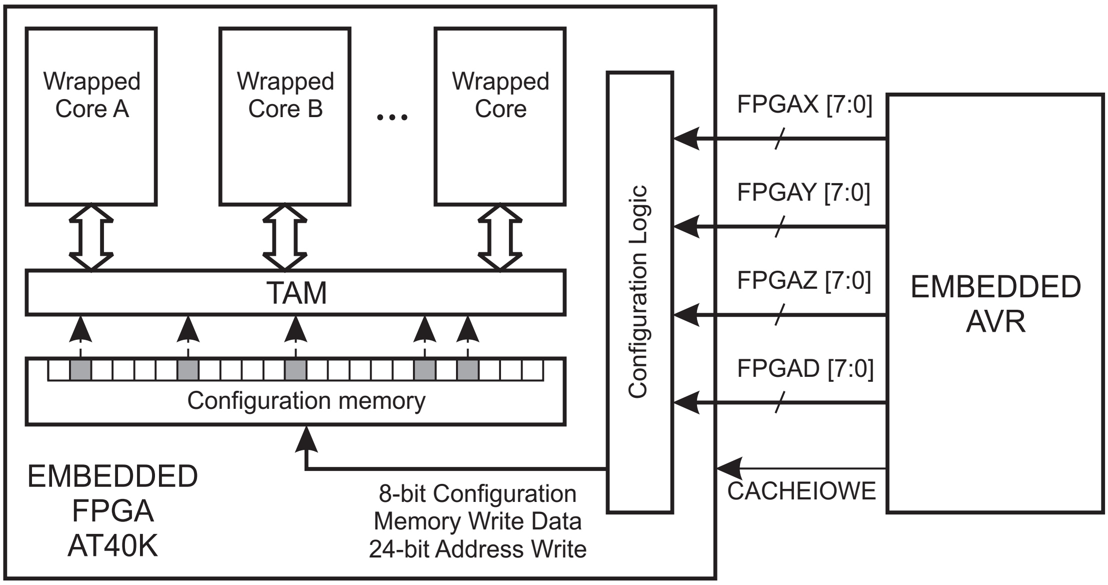

Fig. 4 Atmel FPSLIC TAM configuration scheme

The FPSLIC circuit is connected to PC through JTAG interface. A user is able to program both main parts of IC – program for AVR processor and/or static content of FPGA. Testing with the RESPIN architecture requires reconfiguring circuit cores several times during the test. Each core in the SoC is surrounded by the structure called wrapper [22]. The wrapper allows to connect the core with the defined surrounding either in the functional mode or in the test mode. The Test Access Mechanism (TAM) takes care of the on-chip test pattern transport. The test access mechanism together with wrappers forms the infrastructure for access to individual cores providing tests of all cores. Whereas the core wrapper is defined and standardized by the IEEE 1500 standard, the design of test access mechanism is excluded from this standard and assumed to be addressed by the SoC designer.

An obvious TAM architecture uses multiplexers for reconfiguring ETC and CUT diagnostic data paths connections (see Fig. 5). Every CUT chain input should allow connection with every ETC scan output and every CUT scan output should be connected with the dedicated input terminal of the sink. Every ETC chain output should allow connection with the input of the first ETC chain input through the additional multiplexer which closes the feedback tap. The multiplexer is controlled by the instruction register of the TAM, which is handled by the Wrapper Serial Control (WSC) signal. In case of multiple embedded cores the multiplexers have to switch between corresponding test modes. This approach leads to high area cost for connecting and multiplexing of all core terminals and the control circuitry for these multiplexers will grow substantially with growing number of cores on the chip. Test system requires only one input and one output data channel. It can be easily compared with well known test method called "Boundary Scan" (hereafter BS). BS similarly uses one input and one output data channel and can be applied in SoC for core testing.

Partial FPGA reconfiguration seems to be an efficient way how to form the low area-cost TAM for multiple embedded core SoC design. The FPGA consists of number of generic cells called LUTs. In our system the LUT is used for connecting the test core terminal and a LUT of the TAM. By this arrangement two LUTs are needed to form one wire interconnection between 1-bit core test input and output terminal in the FPGA.

TAM size obviously depends on the number of cores in the SoC and the number of parallel chains of every core. Two TAM architectures, the TAM architecture with multiplexed access and partially reconfigurable TAM architecture, were designed, resulting hardware overheads of the TAMs are compared in Fig. 6. We considered two ETCs for test pattern decompression and the diagram on the left side plots the number of LUTs against the number of cores in the considered SoC with two parallel test chains in every core. The diagram on the right hand side plots the numbers of LUTs against the number of parallel test chains in the ETCs and the CUT, three cores in the SoC are considered. In both cases the TAM using dynamic reconfigurable feature of the FPGA has much less hardware overhead than the multiplexer based TAM.

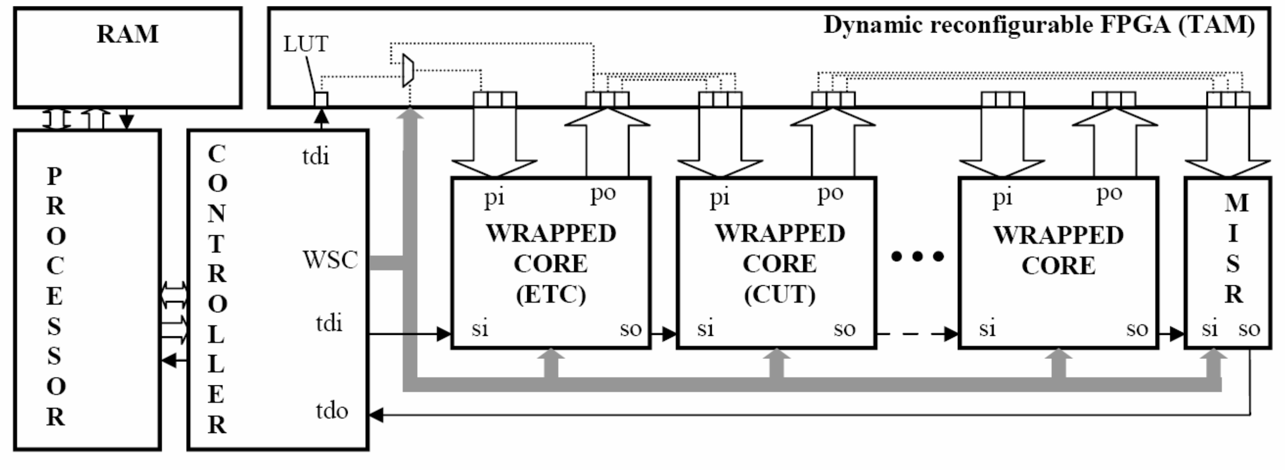

Fig. 5 An example of TAM configuration (given by dotted lines).

The diagnostic system uses an 8-bit AVR processor, an SRAM memory and a dynamic reconfigurable FPGA accessible both from the processor and from the FPGA. In the FPGA we programmed wrapped cores, the MISR, the controller and detached area of the TAM. The AVR processor was used for data processing, for handling the data with the hardware controller and for partial reconfiguration of the TAM before initiation of the core test. Test patterns together with TAM configurations were stored in the embedded SRAM. The processor controls the test scheduling and communicates with the hardware controller. The RAM is used for storing the compressed test sequence. For each test pattern the processor gives the controller a command to run the test cycle independently on the processor. This arrangement enables the hardware controller and the processor to work concurrently and to speed up the test. The hardware controller drives core wrappers and the TAM by signals WSC. During the test cycle the AVR transports one test bit from the memory to the port tdi and informs the controller about availability and suitability of test data (Fig. 1). At the end of the test session, the processor shifts data through the port tdo from the MISR where the responses were accumulated and compares the resulting signature with the sample one from the RAM. After finishing the first CUT test the TAM is partially reconfigured and the next core is assigned as a CUT and it is tested through a newly reconfigured ETC. As the granularity of configurable blocks of the FPGA is relatively fine only a small part of the configuration memory has to be replaced by a new content (In Fig. 4 denoted by the gray color).

Three ISCAS benchmark circuits (S298, S382 and S444) were used as cores in the experiment. The hardware of the diagnostic system including these cores was designed and represented 34% of the AT94K40 resources. For three cores S1423 contained in the SoC 73% of the FPGA AT94K40 resources were exploited. Reconfiguration takes several thousands of clock cycles of processor. Number of clock cycles depends on the design to be reconfigured. In our case the reconfiguration time is less than 1 ms in case of 4 MHz processor clock. The circuit has 36 Kbytes of available RAM memory (20 – 32 Kbytes for program and 4 – 16 Kbytes for data). The size of one reconfigurable bitstream, which was used in the diagnostic system, was 2 Kbytes. The more cores are used in RESPIN architecture the more reconfigurable bitstreams are needed for arranging the ETC–CUT structure. Nevertheless the spent RAM memory amount was acceptable. In case of lack of the RAM memory the bitstreams can be reloaded from a PC. The test time depends on the longest parallel chain and on the number of bits of the compressed test. In our case the test time is about 0.3 ms for the best possible clock frequency of the FPGA (40 MHz).

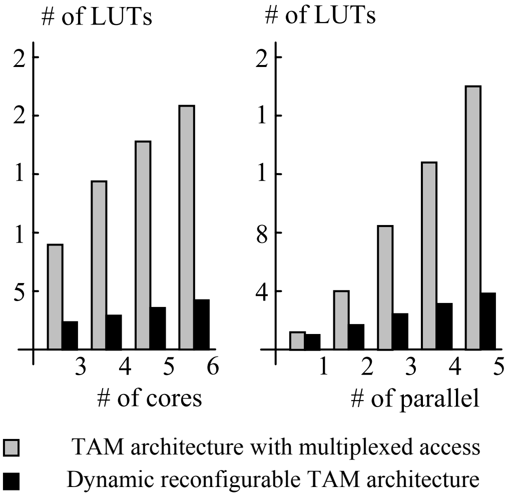

Fig. 6 Comparison of the number of LUTs needed for switching the TAM from one ETC/CUT configuration to another

4. CONCLUSION

The proposed diagnostic system uses highly compressed test patterns; according to our knowledge the compression ratio is better than for other comparable methods. The compression consists of test pattern overlapping. The overlapped patter sequence can be obtained by the COMPAS software tool. For the use of the compressed test sequence in the multi scan chain system the sequence is reordered in order to be correctly decompressed within the RESPIN architecture. We have solved the problem of long CPU time for enumerating the compressed test sequence by multiple usage of test bit usability evaluation during the process of finding the test sequence and by skipping pattern recalculation for cases when don't care bit groups are present in the patterns. This was enabled by using a concatenated list of pattern pointers. Further speeding up of the algorithm could be obtained by using parallel computing.

We have verified that the proposed diagnostic system is applicable on a SoC. We have placed the system together with simple functional cores on the AT94K FPSLIC circuit. The diagnostic system uses the dynamic and partial reconfiguration feature of the embedded FPGA. This is advantageous because it saves resources of the FPGA devoted for switching the TAM busses. For larger cores the system can be built on the large Xilinx FPGA circuits with embedded processor and RAM memory block. This will be the future work of the research team. The property of dynamic reconfiguration of the FPGA part could be an advantage that can save the FPGA resources. The proposed diagnostic system seems to be well suited for SoC built on the future FPGAs with embedded processors and large memories with fine grained dynamical reconfiguration capability and big numbers of LUTs.

We can conclude that the diagnostic system is well suited for a SoC architecture with a processor, RAM block embedded FPGA and ASIC. The memory requirements for storing the test data are lower than it is in case of other comparable methods; the test time is very low, too.

ACKNOWLEDGMENT

The research was supported by the research grant of the grant IQS108040510 of the Czech Academy of Sciences.

REFERENCES

[1] Abhijit Jas, Jayabrata Ghos-Dastir, and Nur A. Touba: Scan Vector Compression/Decompression Using Statistical Coding. Proc. VTS 1999.

[2] http://www.atmel.com/dyn/resources/prod_documents/2818s.pdf [cit 20.5.2006].

[3] Bayraktaroglu, I., and Orailoglu, A.: Decompression Hardware Determination for Test Volume and Time Reduction through Unified Test Pattern Compaction and Compression. Proc. VTS 2003.

[4] Bernhart et al.: OPMISR: the foundation for compressed ATPG vectors. Proc. ITC, 2001, pp. 748-757.

[5] Brglez, F., Bryan, D., Kozminski, K.: Combinational Profiles of Sequential Benchmark Circuits. Proc. Int. Symposium on Circuits and Systems, 1989, pp. 1929-1934.

[6] C Chandra, A. – Chakrabarty, K.: Frequency/Directed Run Length (FDR) Codes with Application to System/on/Chip Test Data Compression. Proc. VTS 2001, pp. 42-47.

[7] http://www.cerc.utexas.edu/itc99-benchmarks/bench.html [cit 20.5.2006].

[8] Chakrabarty, K., Iyengar, V., Chandra, A.: Test Resource Partitioning for System-on-a-chip. Kluwer Academic Publishers, 2002.

[9] Daehn, W., Mucha, J,: Hardware Test Pattern Generation for Built-in Testing. Proc. ITC, 1981, pp. 110-113.

[10] Dorsch, R. and Wunderlich, H-J:Reusing Scan Chains for Test Pattern Decompression. Proc. IEEE ETW, 2001, pp.24-32.

[11] Hellebrand, S., - Liang, H.G. – Wunderlich, H.J.: A mixed mode BIST scheme based on reseeding of folding counters. Proc. ITC, 2000.

[12] IEEE Computer Society. IEEE Standard Test Access Port and Boundary-Scan Architecture - IEEE Std 1149.1-1990. IEEE, New York, 1990.

[13] IEEE Computer Society. IEEE Standard Testability Method for Embedded Core-based Integrated Circuits - IEEE Std 1500-2005. IEEE, New York, 2005.

[14] http://iko.kes.vslib.cz, [cit 10.27.2006].

[15] Immaneni, V., Raman, S.: Direct Access Test Scheme - Design of Block and Core Cells for Embedded ASICs. Proc. ITC, 1990, pp 488–492.

[16] Krishna, C.V., Touba, N.A.: Reducing Test Data Volume Using LFSR Reseeding with Seed Compression. Proc. ITC 2002, pp321-330.

[17] Koenemann, B.: LFSR – coded test patterns for scan designs. Proc. ETC, Munich , Germany, 1991.

[18] Lee H. K., and Ha, D. S.: On the generation of test patterns for combinational circuits. Technical Report 12_93, Department of Electrical Eng., Virginia Polytechnic Institute and State University

[19] Lee H. K., and Ha, D. S.: HOPE: An efficient parallel fault simulator. Proc. DAC, pp. 336-340, June 1992.

[20] Marinissen, E. J.- Zorian, Y. - Kapur, R. Taylor T., and Whetsel. L.:Towards a Standard for Embedded Core Test: An Example. Proc. ITC , pp. 616–627. IEEE, 1999.

[21] Marinissen, E.J., Arendsen, R., Bos, G.: A Structured and Scalable Mechanism for Test Access to Embedded Reusable Cores. Proc. ITC, 1998.

[22] Marinissen, E.J., Goel, S.K., Lousberg, M.: Wrapper design for embedded core test. Proc. ITC, 2000, pp. 911–920.

[23] Novák, O. - Nosek, J.: Test Pattern Decompression Using a Scan Chain, Proc. IEEE International Symposium on Defects and Fault Tolerance in VLSI Systems 2001, pp. 110 – 115.

[24] Novák, O. – Zahrádka, J.: COMPAS – Compressed Test Pattern Sequencer for Scan Based Circuits, Proc. EDCC 2005.

[25] Novak, O., Zahradka, J., Pliva, Z.: COMPAS – Compressed Test Pattern Sequencer for Scan Based

Circuits. Springer: Lecture Notes in Computer Science 3463, ISSN 0302-9743, 2005, pp. 403-414.

[26] Pandey, A. R. – Patel, H. J.: Reconfiguration Technique for Reducing Test Time and Test Data Volume in Illinois Scan Architecture Based Designs. Proc. VLSI Test Symp, 2002, pp. 9-15.

[27] Rao, W., Oraiologlu, A.: Virtual Compression through Test Vector Stitching for Scan Based Designs. Proc. DATE 2003.

[28] Rajski, J. et al.: Embedded Deterministic Test . IEEE Trans. on CAD, vol. 23, No. 5, May 2004, pp. 776-792.

[29] Scandaliaris, J., Moreno, J.M., Cabestany, J., Buttel, P., Rachet, A., Kadlec, J., Hermanek, A., de Saint Romain, D., Habay, G., Donati, A.: A General Design Flow for Dynamically Reconfigurable FPGAs (D_FPGAs). http://www.reconf.org/Files/Publications/RAW03_UPC.pdf [cit 22. 5. 2006]

[30] Schafer, L., Dorsch, R., Wunderlich, H.J.: RESPIN++ Deterministic Embedded Test. Proc. ETW, 2002, pp. 37-42,

[31] Su, C., and Hwang, K.: A Serial Scan Test Vector Compression Methodology. Proc. ITC 1993, PP. 981-988

[32] http://direct.xilinx.com/bvdocs/userguides/ug070.pdf. [cit 9.5.2006]

Received Jun 6, 2007, accepted January 22, 2008

BIOGRAPHIES

Ondřej Novák was born on 1955. Degrees and Titles: 1980 MSc (technical cybernetics), 1987 PhD (computer engineering), thesis on Built-In Test pattern generators, 1993 Associate Professor (computer engineering) (all above mentioned titles from the Czech Technical University in Prague), 2002 full professor at the Technical University in Liberec, Professional career: From 1980 to 1989 with Research Institute for Mathematical Machines, Prague as research worker and senior research worker in the field of micro diagnostics. From 1989 to 2005 he was with the Technical University of Liberec, from 2005 is with the Czech Technical University in Prague, Faculty of Electrical Engineering, Department of computers. He leads the MSc. courses on digital electronics, programmable electronic devices and diagnostics and reliability. His research interests are Easy testable design of integrated circuits, design of Built-In Self-test Equipment, Test pattern generation and compression.

Zdeněk Plíva was born on 22. 2. 1961, in 1984 he graduated (Ing.) at the dept. Automated Control Systems of the Faculty of Mechanical Engeneering at VŠST Liberec, in 2002 he defended PhD. (technical cybernetics) and in 2005 associate professor (technical cybernetics) – both at Technical University of Liberec, Faculty of mechatronics. In 1984–1989 he was with Elitex Chrastava (technician designer), in 1989–1997 with Elitron Liberec (dept of PCB). Since 1997 he is with Technical University of Liberec; his research interests are diagnostics of ICs, IC design, PCB manufacturing and assembly.

Jiří Jeníček was born on 16. 12. 1978. In 2005 he graduated (MSc.) at the Faculty of Mechatronics and Interdisciplinary Engineering Studies at Technical University of Liberec, department of Electronics and Signal Processing. Since 2005 he is studying for PhD. in the field of test data compression. His scientific research is focusing on test data compression using pattern overlapping.

Zbyněk Mader was born on 5. 3. 1972. In 1996 he graduated (MSc.) at the department of Telecommunication Engineering of the Faculty of Electrical Engineering at Czech Technical University in Prague. He defended his PhD. in the field of testing system on a chip in 2007; his thesis title was „Core Based SoC Diagnostic System with Reduced Memory Requirements". Since 2003 he is working as a professor assistant with the Department of Electronics and Signal Processing. His scientific research is focusing on Diagnostics and Reliability.

Michal Jarkovský was born on 28.4.1979. In 2004 he graduated (MSc.) at the Department of Electrical Engineering of the Faculty of Mechatronics and Interdisciplinary Engineering Studies at Technical University of Liberec. He is Ph.D. student at the Institute of Information Technology and Electronics in the area of Technical Cybernetics. His research focuses on Boundary-Scan testing and diagnostics of a SOC, specially testing analog and mixed-signal integrated circuit.